



**CONGESTED HIGHWAYS ACTION RESPONSE TEAM  
STATE HIGHWAY ADMINISTRATION**

---

## **CORBA ORB Evaluation**

**For CHART II**

**Contract DBM-9713-NMS  
TSR # 9901961  
Document # M361-AR-004R0**

**March 19, 1999  
By  
Computer Sciences Corporation and PB Farradyne Inc**



# **CORBA ORB Evaluation**

**For**

## **CHART II**

Rich Doyle

Scott Melby

Sudhakar Nallamothu

Desikan Santhanam

March 19, 1999

## Table Of Contents

<b>OVERVIEW .....</b>	<b>1</b>
<b>EVALUATION CRITERIA.....</b>	<b>2</b>
<b>VENDOR INFORMATION .....</b>	<b>4</b>
IONA TECHNOLOGIES.....	4
INPRISE CORPORATION .....	4
OBJECT ORIENTED CONCEPTS (OOC).....	4
<b>EVALUATION FINDINGS .....</b>	<b>5</b>
CORBA IMPLEMENTATION .....	5
QUALITY .....	5
CUSTOMER SUPPORT .....	5
INTEROPERABILITY .....	6
CORBA SERVICES .....	6
PERFORMANCE .....	7
EASE OF USE .....	7
SOURCE CODE AVAILABILITY .....	8
PRICING.....	8
<b>CONCLUSION .....</b>	<b>10</b>
<b>APPENDIX A – ACRONYMNS.....</b>	<b>11</b>
<b>APPENDIX B – SUMMARY MATRIX.....</b>	<b>12</b>
<b>APPENDIX C – PRICE LIST .....</b>	<b>13</b>
<b>APPENDIX D – REFERENCES.....</b>	<b>15</b>
<b>APPENDIX E – PERFORMANCE RESULTS .....</b>	<b>17</b>
<b>APPENDIX F – INTEROPERABILITY TEST RESULTS.....</b>	<b>21</b>
<b>APPENDIX G - SERVICE AVAILABILITY MATRIX.....</b>	<b>22</b>

# Overview

The CHART II system will be implemented utilizing CORBA, the Common Object Request Broker Architecture. CORBA is an open standard defined by the Object Management Group (OMG) that specifies middleware for distributed object oriented systems. Many vendors implement the CORBA standard and sell their implementation as an Object Request Broker (ORB). This vendor supplied Object Request Broker comes in the form of a code library that is utilized by every application in the system. The vendor supplied ORB handles the CORBA compliant communications involved when invoking operations on distributed objects, thus freeing application developers from writing such code.

In addition to specifications for ORBs, there are also specifications for applications that are commonly needed in distributed systems. There are at least 15 such applications, known as CORBA services, specified with others being added. Many vendors that sell ORBs have also implemented some of the CORBA services. It is possible (and may be necessary) to run one vendor's CORBA service and have it inter-operate with software built using another vendor's ORB.

Since CORBA and CORBA Services are specifications, vendors are given great latitude in how they implement the specifications while still being able to claim they are CORBA compliant. This being the case, ORBs and CORBA services are not commodities in today's market and require evaluation prior to purchase, especially given their price range.

Evaluating an ORB can be a time consuming activity. Given twenty plus vendors on the market, it is not feasible to evaluate each ORB using a "hands on" approach. Instead, time was taken to whittle down the number of vendors using information from vendor web sites, news groups, and print media. This pre-evaluation allowed us to limit our evaluation to three vendors, two of which are market leaders (Iona and Inprise), while another is an up and comer that has been receiving a lot of attention in the software industry (Object Oriented Concepts).

This paper discusses the results of the evaluation of the CORBA products provided by these vendors and provides a recommendation of the vendor whose ORB should be used for the implementation of the Chart II system.

# Evaluation Criteria

While evaluating ORBs, there were a number of both tangible and intangible criteria we used to differentiate between each vendor's products. These criteria are not listed in order of importance or significance. Where appropriate the importance of each criterion is discussed along with an explanation of why the team believed this should be the case.

<b>CORBA Implementation</b>	Since the CORBA specification does not govern implementation, we looked at how each vendor accomplished meeting the specification. For example, some vendors require vendor specific "helper" applications to be deployed in the system to allow the "real" CORBA applications to function.
<b>Quality</b>	Our evaluation goal was to determine if any of the vendor's products were buggy.
<b>Customer Support</b>	Our evaluation goal was to determine how timely we could get support for questions and problem resolution.
<b>Interoperability</b>	Our evaluation goal was to make sure that objects within the CHART II system would be able to be accessed with CORBA software that uses an ORB from a vendor other than the one used in CHART II. Since this type of interoperability is inherent in the CORBA specification, this testing really identifies when a vendor is not compliant to the specification.
<b>Performance</b>	The performance of the ORB can affect the performance of the entire system. The performance testing was not used to make the fastest ORB our choice and instead was used to determine if one ORB was significantly faster or slower than the others.
<b>Ease of Use</b>	Our evaluation goal was to determine which ORB was easiest for programmers to use with regard to compiling, debugging, and deployment. This is not a quantifiable measurement, but we wanted to identify any problem areas that were encountered during our evaluation.
<b>CORBA Services</b>	Our evaluation goal was to determine which CORBA Services were available from each vendor. This information does not weigh heavily on our decision due to the fact that services are interoperable and do not need to be purchased from the same vendor that supplies your ORB.
<b>Vendor Longevity</b>	Our evaluation goal was to determine if any of the vendors seemed likely to go out of business.
<b>Source Code Availability</b>	Although you would not want to change the source code for your ORB, having the source code provides a number of advantages. It allows complete application debugging and also provides a safety net in case the ORB vendor should go out of business or stop supporting their CORBA products (such as Hewlett Packard or Sun Microsystems).

**Price**

There is a great variance in the price of ORBs, even though each one implements the exact same specification. During our evaluation, price was considered as a secondary factor with more focus being placed on the products themselves.

# **Vendor Information**

## **Iona Technologies**

Iona is headquartered in Dublin, Ireland with US headquarters in Cambridge, MA. Iona has over 550 employees and their 1998 annual revenue was over \$58 million. Iona has over 3500 customers, including Boeing, BellSouth, Motorola, Chevron, and Ericsson. Iona stock is traded publicly on NASDAQ and the Irish stock exchange. Iona's Orbix ORB has been a market leading product for the past few years.

## **Inprise Corporation**

Inprise is headquartered in Scotts Valley, CA with corporate offices worldwide. The company was formerly known as Borland International. Borland purchased Visigenics in 1997, bringing with them the Visibroker ORB.

Inprise reported revenues of \$189 million in 1998. Inprise's Visibroker ORB is a market leader with many customers including First Union Bank and the American Automobile Association (AAA). Visibroker is also integrated with several products, including the Netscape Navigator.

## **Object Oriented Concepts (OOC)**

OOC is a small growing company with headquarters in Billerica, MA and offices in Ettlingen, Germany and expanding to Canada. OOC was founded in 1996, with the first release of their ORB (Omnibroker) occurring in Q1/1997. The ORB product name has since been changed to ORBacus.

The vendor has many satisfied customers using its ORB, including Hewlett Packard, NEC, MCI/UUNET, Lockheed Martin, and CSC.

# Evaluation Findings

## CORBA IMPLEMENTATION

While all three vendors implement the CORBA specification, ORBacus is the only implementation that does not require a proprietary server to be running on the network for the ORB to function properly. Orbix requires that an Orbix Deamon be running on the network to allow CORBA applications to run. Visibroker requires its Smart Agent to be running on the network.

While requiring a proprietary server application to be running does not affect the openness of the system, it does create complexity in administration.

## QUALITY

During our evaluation, only one vendor's quality really came into question. The Orbix evaluation disk would not install properly on two out of three machines. After receiving instructions for a work around from Iona technical support, the install process still crashed. This issue delayed the evaluation for the Orbix product for several days and was never satisfactorily resolved. Our developers were able to play around with values in the registry to finally allow the software to be installed properly. The Orbix ORB also failed on the interoperability tests that we performed. Iona technical support suggested a work-around for this problem, but this did not fix the problem. Iona technical support also suggested that we download a patch for the product, but the patch was not acquired prior to this writing.

## CUSTOMER SUPPORT

It is the belief of the team that customer support is a critical factor for determining the quality of an ORB product. We base this belief on the cutting-edge nature of CORBA technology, the complexity of CORBA ORB implementations, and the fundamental necessity of having a robust and reliable ORB to building a reliable distributed software system.

During our evaluation, we made contact with each of the ORB vendors for technical support. The details of each of these support encounters is described below.

OOC provided the best support of any of the vendors. While the support was given via e-mail, a response was always received within 5 minutes. The support received from OOC is not limited product support. OOC employees are always willing to discuss the CORBA specification, future directions of CORBA, or application design issues. OOC also has a list server, allowing one to enlist the help of other ORBacus users. Questions posted to the list often elicit responses from well respected CORBA consultants such as Michi Henning, author of "Advanced CORBA programming with C++".



The support received from Iona was poor in one instance and little better in another. Problems downloading a file from their FTP server prompted an e-mail to the person in charge of download problems. A response was never received. A call to technical support regarding an install problem was answered the same day, however the solution given did not work. A second contact was made to Iona for support on this same issue. Once again a possible fix was suggested but the application of the fix still did not leave us with a working installation of their product.

An attempt to contact Inprise for Visibroker support during the evaluation was denied due to the fact that we had not yet purchased their product. The evaluation team then turned to Internet news groups for support by posting a question about our issue. An answer to the question that was posted was never answered, however the team discovered a solution to the problem on our own.

## **INTEROPERABILITY**

The goal of the interoperability tests was to verify that each ORB vendor was CORBA IIOP compliant. Compliance to the IIOP specification guarantees that an application written based on a particular ORB vendor's product will be able to communicate with another application program implemented based on any other IIOP compliant ORB product given only an object reference and, optionally, the IDL which specifies the object interface.

Two separate types of tests were conducted. The first was informal communication testing. As we ported our server and client applications to the different ORB products, we tested them out against existing servers which were already running. Our findings from these tests were that Visibroker and ORBacus were highly interoperable. We were able to port the Visibroker server application and communicate with the ORBacus client. At a later phase in the evaluation we were also able to port the Visibroker client and communicate with our ORBacus server. The installation problems encountered with the Orbix product precluded us from evaluating their ORB in this manner.

Our second set of interoperability tests were more rigid. The throughput test server and client applications were ported to each ORB. Each permutation of client and server was then tested for interoperability. The results of these tests can be found in *Appendix F – Interoperability Test Results*.

Additional interoperability tests were performed on the Orbacus ORB to ensure that it was interoperable between the Unix and Windows NT operating systems. This test involved running a server implemented using Orbacus for Windows NT and a client implemented using Orbacus/Java. In this test the server was run at PBF Rockville and the client was run from the CHART lab at CSC Hanover on a Sun Sparcstation running the Solaris operating system. All aspects of this test were successful.

## **CORBA SERVICES**

CORBA services are application programs that address common needs of all distributed architecture applications. The Object Management Group (OMG) provides a specification for

the interfaces that a particular service must meet in order to be considered CORBA compliant. Although every service must fulfill the interface as specified, thus making them somewhat of a commodity item, there are some issues which make it desirable to use the services provided by the ORB vendor. These issues include ease of communicating with the service, ease of maintaining the service, and avoidance of potential interoperability issues.

The evaluation team designed a prototype application architecture which we felt would strongly resemble a basic real-time ITS command and control system. This architecture relied on the availability of a CORBA Trading service for object discovery and a CORBA Event service for push update-on-change capabilities. A separate paper entitled “**Object Request Broker Evaluation Prototype**” provides details regarding the design of this prototype.

OOC provided evaluation copies of their CORBA Trading service and CORBA event service. The only service related problem we had was an overflow of events in the event service which caused some of the push updates to be dropped. This problem was due to a default queue size in the event service that was easily overridden with a configuration file change.

Inprise Visibroker does not have a CORBA trading service; they use their proprietary Smart Agent executable for object discovery instead. The Orbix Trading service was not obtained in time to be included in this evaluation. However, the team had no trouble using the ORBacus Trading service with these other ORB products and recommends its use on this project independent of which ORB is used.

The only other CORBA service worth mentioning at this time is the CORBA Security service. The development team is currently determining if this service will be useful in meeting the security requirements of the CHART II project. None of the vendors provided an evaluation security service. At the time of this writing only Iona has an implementation of the security service available for purchase.

Refer to Appendix G - Service Availability Matrix for a matrix of CORBA Service availability by vendor.

## PERFORMANCE

In order to test the performance of each of the ORB products, the development team implemented a throughput test which tracked the round trip time of function calls from a test client application to a server. For complete details on the function calls made and the results of these tests please refer to **Appendix E – Performance Results**. The function calls tested the speed with which the ORB handled remote server calls for each of the CORBA basic types and arrays of the basic types. The high-level finding of these tests was that ORBacus and Visibroker were far more efficient than Orbix.

## EASE OF USE

The development team found each of the vendor’s products usable from the perspective that we were able to get a distributed application up and running on each ORB. The team felt that ORBacus from OOC was the most usable of the products due to their very tight adherence to

the CORBA 2.x specification, lack of proprietary extensions, active list server, and incredible technical support.

The Visibroker product was also usable due largely to their excellent documentation package which includes a full html document available at their web site [www.inprise.com](http://www.inprise.com). However, the team found their requirement that a proprietary Smart Agent service be running on the local area network detracted from usability. Additionally, the Visibroker ORB library used the old iostream library which has been supplanted by the improved standard C++ library iostream implementation. There was no easy work-around for this issue because the team did not have the product source code. Thus we were forced to retrofit our application code to use the old stream library.

Iona's Orbix was the least usable of the products due to an installation bug, inadequate technical support, and the proprietary Orbix Daemon.

Both Orbix and Visibroker utilize Dynamic Link Libraries for their ORB implementations. This complicates deployment of an application built using these products because it requires that an installation procedure be run on each computer on which the applications will be deployed.

## **SOURCE CODE AVAILABILITY**

OOC is the only vendor that provides source code for their ORB. This allows full debugging of CORBA applications. This is important because it is possible for application code to cause the ORB code to throw an exception. In such an instance, having the source code for the ORB allows the programmer to see what is failing in the ORB and trace it backwards to the application code. Without the source code, the possibilities for the cause of an exception are endless, and debugging the problem can prove very difficult.

Another benefit of having the source code is the ability to apply patches prior to an official release. This is useful if bugs are found. Instead of having to wait for the vendor to package a release, the vendor will tell which code module to patch and supply the source code to change. You can then update your own copy of the ORB for a temporary fix and apply the official release later. This type of support was provided to the development team via the OOC list server on one occasion.

Lastly, having the source code provides a security blanket in case a company goes out of business or stops supporting the product. This has happened with Sun Microsystems' ORB as well as Hewlett Packard's ORB, so the size of the company does not necessarily guarantee product longevity.

## **PRICING**

The pricing of ORBs differs greatly among vendors. Orbix is the most expensive, while ORBacus is the least expensive. For a sample system involving 1 server machine and 11 clients, an Orbix deployment would cost \$63,820 dollars more than an ORBacus deployment.

A major pricing area where the vendors differ is in run-time licenses. Orbix and Visibroker both have a run-time fee near \$2000 dollars, while ORBacus has no run-time fee. This means that as more servers and clients are added to the system, the Orbix or Visibroker deployment cost will climb.

The initial pricing difference along with the cost of system expansion makes ORBacus a clear leader in pricing. This pricing combined with the functionality and quality of the ORB makes ORBacus a good value. See Appendix C – Price List for more details.

## Conclusion

After extensive evaluation of each of the ORB products and much monitoring of industry opinion, it is our conclusion that the ORBacus ORB provided by OOC is the best choice for the CHART II project. We base this conclusion on OOC's excellent product quality, tight adherence to the CORBA specification, and above all, their commitment to customer service. We believe that the potential risk of OOC going out of business and discontinuing support for their product is mitigated by their adherence to the CORBA specification and their open-source model of product deployment. It is also our conclusion that, due to ORBacus' interoperability with other vendor products, we will be able to incorporate CORBA services provided by other vendors should the need arise. Additionally, if the goal is to build a system which will eventually become a cornerstone for an interstate interoperable ITS monitoring system, we believe that choosing the vendor who most closely matches the CORBA specification goes a long way toward that end.

## **Appendix A – Acronymns**

BOA	Basic Object Adapter
CORBA	Common Object Request Broker Architecture
IIOP	Internet Inter-ORB Protocol
IOR	Interoperable Object Reference
OMG	Object Management Group
OOC	Object Oriented Concepts
POA	Portable Object Adapter

## Appendix B – Summary Matrix

The following table shows a rating for each vendor in each of our evaluation categories. Five stars indicates the best possible rating, while no stars represents the worst possible rating.

Object Request Brokers			
Vendor	OOC	IONA	Inprise
Product	ORBacus	Orbix	Visibroker
Product Evaluation Criteria			
<b><i>CORBA Implementation</i></b>	* * * * *	* * *	* * *
<b><i>Quality</i></b>	* * * *	* *	* * * *
<b><i>Customer Support</i></b>	* * * * *	* * *	*
<b><i>Interoperability</i></b>	* * * * *	patch needed	* * * * *
<b><i>Performance</i></b>	* * * * *	* * *	* * * * *
<b><i>Ease of Use</i></b>	* * * *	* * *	* * *
<b><i>CORBA Services</i></b>	* * * *	* * * * *	* *
<b><i>Vendor Longevity</i></b>	* * *	* * * * *	* * * * *
<b><i>Source Code Available</i></b>	* * * * *	no	no
<b><i>Price Competitiveness</i></b>	* * * * *	* *	* *
<b>Overall Product Rating</b>	* * * * *	* * *	* * * *

## Appendix C – Price List

<b><i>Product</i></b>	<b><i>Iona / Orbix</i></b>	<b><i>Inprise / Visibroker</i></b>	<b><i>OOC / ORBacus</i></b>
ORB – C++ Developer (First)	\$5330	\$1995	\$4290
ORB – C++ Developer (Subsequent)	\$5330	\$1995	\$2490
ORB – Java Developer (First)	\$1499	\$1995	\$995
ORB – Java Developer (Subsequent)	\$1499	\$1995	\$595
ORB – Server Deployment	\$6150	\$1895 per processor	\$0
ORB – Client Deployment	\$2050	\$1895 per processor	\$0
Trading Service Deployment (First)	Price not available	Not Available	\$3495
Trading Service Deployment (Subsequent)	Price not available	Not Available	\$1995
Event Service Deployment	Price not available	\$0	\$0

### Sample Total Cost

The following sample shows the difference in total cost for CORBA products for a system that may require product volumes similar to CHART II. This sample is used to magnify the price differences that are not easily seen when looking at itemized pricing as shown above. These numbers do not in any way represent the actual price for Chart II and should not be construed as a quote or cost estimate.



Product	Quantity	Iona / Orbix	Inprise / Visibroker	OOC / ORBacus
C++ Developer License	13	\$69,290	\$25,935	\$34,170 (\$ 8,135 Java)
Deployment License for Dual Processor Server	1	\$6,150	\$3,790	\$0
Deployment License for single processor Client	11	\$22,550	\$20,845	\$0
Trading Service	1	\$3,495 (use ORBacus Trader)	\$3,495 (use ORBacus Trader)	\$3,495
Event Service	1	\$0	\$0	\$0
TOTAL		\$101,485	\$54,065	\$37,665 (\$11,630 Java)

## Appendix D – References

Following are e-mails that were posted to various news groups, such as comp.object.corba, where developers discuss topics related to CORBA. (Names have been removed)

The CORBA solution meets all the client requirements and provides a solid foundation for a platform-independent distributed object architecture. It also saves an enormous amount of development time.

Having settled on a CORBA solution, we began a comparison of CORBA vendors. We found that one company, OOC, provides a product that best suited our needs. OOC's ORBacus:

- Is less expensive
- Has no royalty charges for deployment
- Provides the source code
- Is fully CORBA compliant

We have been extremely pleased with both ORBacus and OOC. OOC provides excellent technical support and they constantly improve the product and provide new releases. We have found that ORBacus provides us with a reliable and robust implementation.

---

> Hi,  
>  
> This probably isn't the correct forum for this  
> question but I'm trying to avoid a marketing type response.  
>  
> There are about 30 orbs available these days that  
> I've reviewed. While many seem very good, only a few are  
> commercially viable (useable in a commercial product) IMO:  
> Orbix, Visibroker, and Orbacus, (maybe MICO and OmniOrb).  
>  
> I have direct experience with why I don't want to use Orbix anymore.  
>  
> However, why should I choose Orbacus over Visibroker?

One word - support, support support! By the way, did I mention support?

I've been hacking (er, developing, ahem!) software for over 25 years, and my experience working with the OOC staff is by far the most collaborative, cooperative, and responsive experience I have ever had with a vendor. I don't have experience with Visibroker, but I tried to work with Orbix for almost two years before I threw up my hands in frustration at their poor product quality and egregious support. Since switching to Orbacus, we no longer have to endure such nonsense. The folks at OOC could not be more helpful, knowledgeable, and just down right professional about matters-CORBA. While Orbacus might not have every bell and whistle that CORBA specifies (yet!), they definitely have the essential stuff (and maybe more than I am giving them credit for), and by golly, it works! And, to a much larger degree than the likes of Orbix, in high compliance with the

latest specifications.

---

>Having been subjected to Orbix for more than a year, I'll have to disagree.  
>Visi is a much easier product to use compared to Orbix. We used both  
>on the project I was on, and Visi gave us very few problems, whereas Orbix  
>was a nightmare. Take for example, the amount of time spent trying to find  
>a version of the products that would even work:

[snip of further stuff I completely agree with!]

I must agree with everything you said, and state that my experience with Orbix was as bad, if not worse from a support perspective. Rather than drag Iona through more public mud, suffice it to say I go out of my way to steer new projects away from that unstable, "unsupported", ORB. If you must use a commercial ORB, check out Visibroker. My personal preferences are the open-source implementations, especially ORBacus, omniORB, and MICO. If you're willing to join a truly open-source mindset, then have a look at ACE TAO!! There's lot's of good open-source CORBA work going around these days. ;)

---

>  
> On Thu, 04 Feb 1999 23:37:45 GMT, <name removed> wrote:  
>  
> >If you must use a commercial ORB, check out Visibroker. My personal  
> >preferences are the open-source implementations, especially ORBacus,  
> >omniORB, and MICO. If you're willing to join a truly open-source mindset,  
> >then have a look at ACE TAO!!  
>  
> <nitpick> OmniORB, mico, TAO, and ILU all have an "Open Source(tm)"  
> development model & license, so saying that one is more open source than  
> the others is hardly fair to the others.  
>

And ORBacus is \*not\* "Open Source (tm)", although they do supply source, their prices are very reasonable, and they have excellent products and support. If the idea of using "public domain" software freaks out your managers then ORBacus is a good choice. Well, it is a good choice anyway...

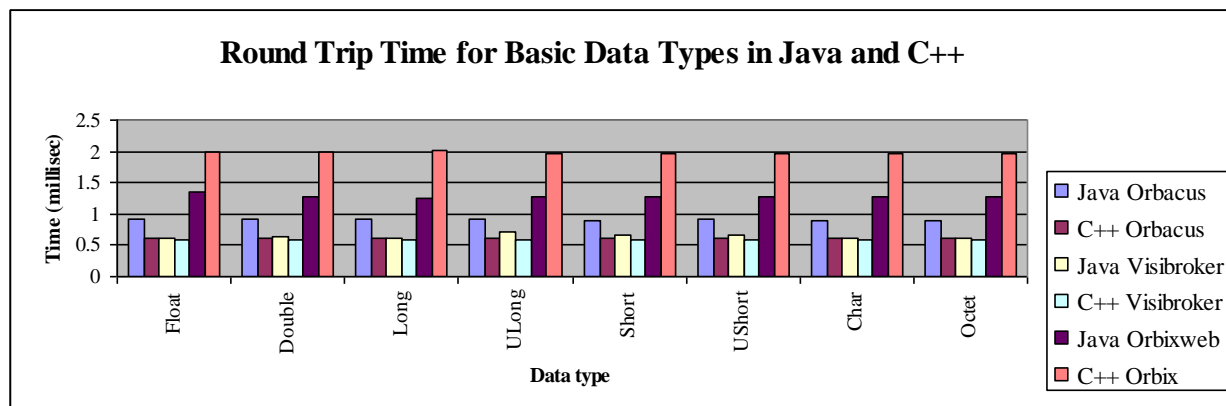
## Appendix E – Performance Results

The objective of this test was to evaluate the speed of the ORB by passing data of different types between a client application and the server, and measure the round trip time taken by the ORB to get a response. In this test, basic data types (integer, real, character, octet, any) were transferred as separate entities, as arrays, and as sequences. The size of the array and sequence used were 1024 bytes. The test was performed by invoking an operation upon a server with each of the data types as an input parameter and measuring the elapsed time between the initiation of the call to the server and the response received from the server. The test was repeated 10000 times for each data type and the time reported is the average of the calls for that particular data type.

**System Configuration:** The systems used for testing were two Pentium 400 MHz, 128 MB RAM, Windows NT4.0 Workstations for server and client connected on a local isolated 10 Mbps Ethernet network.

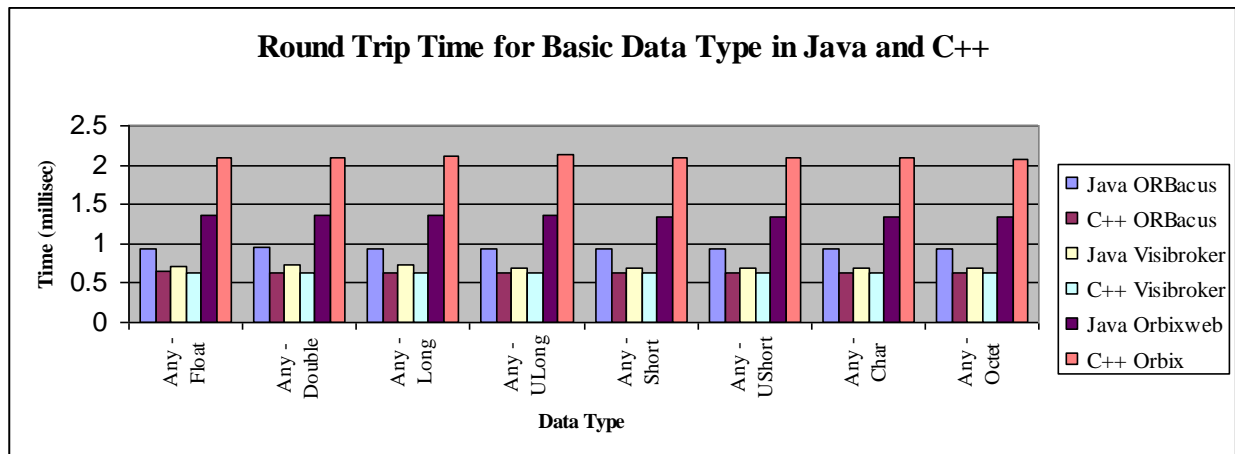
**ORB's tested:** Visibroker, ORBacus, Orbix

### Results



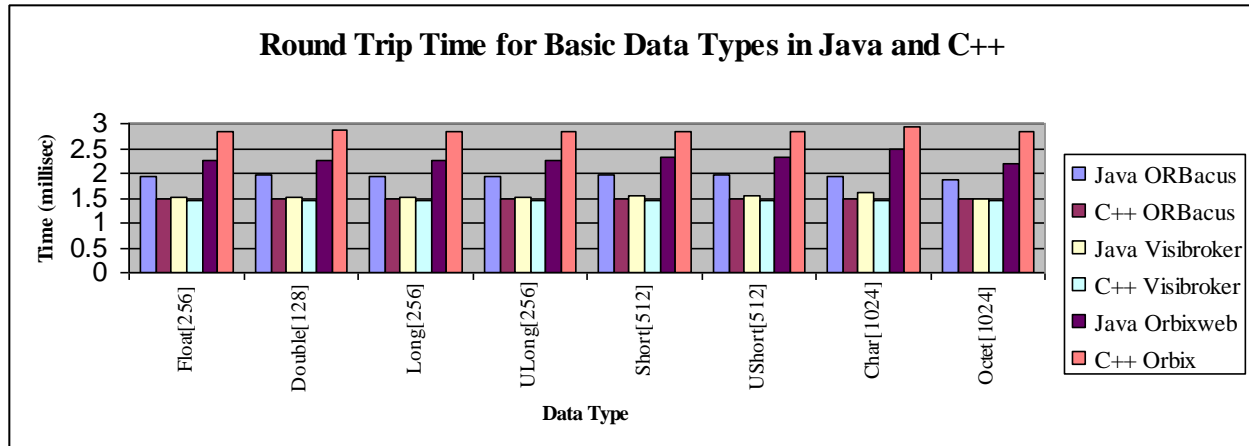
Round trip time in milliseconds

Server →	Java Orbacus	C++ Orbacus	Java Visibroker	C++ Visibroker	Java Orbixweb	C++ Orbix
Float	0.91	0.612	0.619	0.587	1.345	1.984
Double	0.926	0.619	0.63	0.594	1.283	1.986
Long	0.911	0.612	0.619	0.586	1.26	2.026
ULong	0.91	0.611	0.724	0.585	1.269	1.971
Short	0.904	0.615	0.675	0.579	1.264	1.974
UShort	0.91	0.614	0.666	0.596	1.265	1.974
Char	0.9	0.611	0.608	0.584	1.264	1.976
Octet	0.904	0.611	0.615	0.583	1.267	1.975



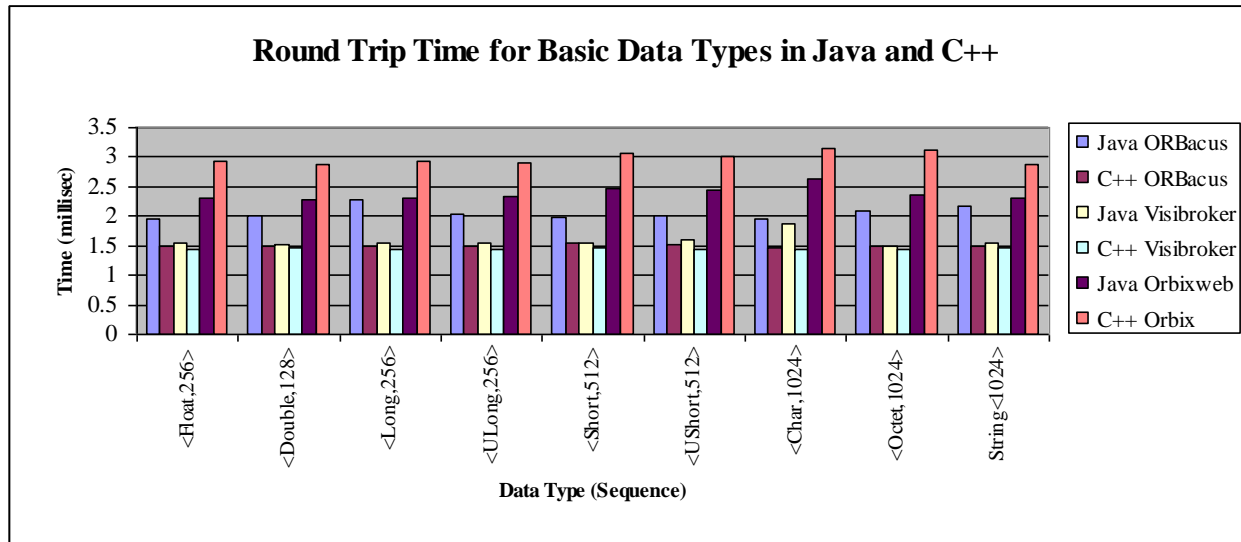
Round trip time in milliseconds

Server →	Java ORBacus	C++ ORBacus	Java Visibroker	C++ Visibroker	Java Orbixweb	C++ Orbix
Any - Float	0.945	0.643	0.703	0.63	1.359	2.09
Any - Double	0.948	0.64	0.737	0.632	1.36	2.092
Any - Long	0.937	0.639	0.73	0.625	1.355	2.122
Any - ULong	0.939	0.638	0.699	0.624	1.361	2.133
Any - Short	0.94	0.631	0.697	0.624	1.345	2.085
Any - UShort	0.938	0.631	0.699	0.625	1.351	2.086
Any - Char	0.939	0.632	0.699	0.625	1.351	2.085
Any - Octet	0.937	0.634	0.698	0.625	1.342	2.083



Round trip time in milliseconds

Server →	Java ORBacus	C++ ORBacus	Java Visibroker	C++ Visibroker	Java Orbixweb	C++ Orbix
Float[256]	1.945	1.475	1.53	1.442	2.263	2.836
Double[128]	1.968	1.475	1.529	1.442	2.264	2.865
Long[256]	1.95	1.473	1.528	1.441	2.266	2.836
ULong[256]	1.945	1.47	1.529	1.442	2.266	2.836
Short[512]	1.972	1.471	1.547	1.44	2.327	2.838
UShort[512]	1.966	1.472	1.55	1.438	2.331	2.839
Char[1024]	1.937	1.47	1.601	1.438	2.469	2.932
Octet[1024]	1.872	1.475	1.491	1.436	2.209	2.843



Round trip time in milliseconds

Server → Sequence ↓	Java ORBacus	C++ ORBacus	Java Visibroker	C++ Visibroker	Java Orbixweb	C++ Orbix
<Float,256>	1.956	1.484	1.542	1.45	2.315	2.928
<Double,128>	2.007	1.483	1.532	1.453	2.291	2.876
<Long,256>	2.272	1.479	1.533	1.449	2.319	2.933
<ULong,256>	2.039	1.487	1.536	1.45	2.32	2.914
<Short,512>	1.97	1.545	1.542	1.452	2.465	3.078
<UShort,512>	2.012	1.52	1.592	1.449	2.43	3.011
<Char,1024>	1.94	1.476	1.864	1.449	2.634	3.146
<Octet,1024>	2.093	1.49	1.489	1.448	2.373	3.109
String<1024>	2.177	1.504	1.554	1.466	2.31	2.865

## Appendix F – Interoperability Test Results

Client → Server ↓	Visibroker	ORBacus	Orbix
Visibroker	OK	OK	Fails to resolve
ORBacus	OK	OK	Fails to resolve
Orbixweb	OK	OK	OK
Orbix	OK	OK	OK

Interoperability problems:

- Orbix 2.3 client could not communicate with either a Visibroker server or an ORBacus server because it failed to resolve the object references. Information available on the CORBA news group suggests that Orbix has a patch to solve this problem.

No other problems were encountered regarding interoperability of the individual brokers.



## Appendix G - Service Availability Matrix

The following table shows the CORBA Services that are implemented by the evaluated vendors. Shaded cells indicate that a vendor has made an implementation of the given CORBA service available for purchase. Since our evaluation did not evaluate many of these services, no claim is to be made as to the compliance or quality of a vendor's implementation.

<b>CORBA Service</b>	<b><i>Iona</i></b>	<b><i>Inprise</i></b>	<b><i>OOC</i></b>
Collection			
Concurrency			
Event (Typed)			
Event (Untyped)			
Externalization			
License			
Life Cycle			
Naming			
Notification			Beta
Persistence			
Property			
Query			
Relationships			
Security			
Time			
Trader			
Transaction			

Note: A security service integrated with ORBacus Java is available from a third party vendor.